CS 181

Project 1: A Computer Cost Comparison Tool

**Due**: Wed. Sep. 9, at the beginning of class

In business, the cost to own a piece of equipment is calculated by adding the following things together:

1. The value that the piece of equipment loses over time, called the depreciation. This is the cost of the piece of equipment when you bought it, minus the money you get when you later sell it. The price you get when you sell it is called the “depreciated value”. The depreciation is calculated as
2. The cost to operate the piece of equipment over a fixed period of time.

Write a program that can be used to analyze the cost of ownership of two different computers over the same number of years. One computer will be a laptop, and the other will be a desktop.

To calculate the depreciated value of an item, you need to know

1. The original cost.
2. The number of years it has been owned.
3. The depreciation rate. The depreciation rate for laptops is 15% per year. This is the amount of value the laptop loses each year. The depreciation rate for desktop systems is 20% per year.

Once you know these values, the depreciated value is calculated by the formula:

The cost to operate a computer is calculated by the number of hours it is used each year, multiplied by the cost of the electricity is uses each hour.

* Desktop systems use $0.015 of electricity each hour.
* Laptop systems use $0.006 of electricity each hour.

Finally, laptop computers require a new battery every three years. The cost of a battery is $200. To calculate battery-replacement cost, assume the battery is replaced once every three-year period. For example, if a laptop is owned for two years, there is no battery replacement cost. If a laptop is owned for seven years, the battery would be replaced twice.

Your program should prompt the user for the following information:

* The name and original cost of a laptop
* The name and original cost of a desktop system
* The number of hours the computers are used each year.
* The number of years the computers are owned.

After prompting and reading the information from the user, calculate and print out the cost of ownership for both computers.

Here is an example of what your program *must* look like when it is executed. User input is shown in **bold**.

Enter name of laptop computer: **MacBook Pro**

Enter cost of laptop computer: **1299.00**

Enter name of desktop computer: **Dell Tower of Awesomeness**

Enter cost of desktop computer: **1100.00**

How many hours will each computer be used in a year? **1750**

How many years will you own each computer? **5**

Cost of owning a Macbook Pro over 5 years is $975.13

Cost of owning a Dell Tower of Awesomeness over 5 years is $870.80

To receive full credit, your program must meet the following specifications.

* It must have two classes: One to store information about laptops, and one to store information about desktop systems. Each class should be split into header and implementation files.
* Each class should have private member variables to store the computer name, cost, hours used per year, number of years used, and total cost over time. The laptop class will need a constant value to store battery replacement cost.
* Write public methods that allow you to manipulate these variables as your program needs to. Each class must have at least one constructor.
* Be sure that the cost of owning each computer over time is calculated by a method in its class. Don’t do this calculation in the main program.
* Comment your code appropriately. Include an initial comment at the top of your main program that includes your name, the date, and a brief description of what the program does. In addition, comment each logical segment of code in your main program, and each method in both classes.
* Declare constants for literal values that have a special meaning in the program (such as battery lifetime, depreciation rate, etc.) Beware of “magic numbers”.
* Use meaningful identifier names - no one letter identifiers or weird abbreviations. Follow C++ capitalization conventions when naming constants, methods, functions, classes, etc.

**What to turn in:** When you have finished with your program, submit all five source files to blackboard.